CONNECTED COMMUNITIES

CONFINE

Initial management guide of the testbed

Deliverable D3.2

Date: 17th September 2013

Version: 1.0

Editor:	Lorena Merino, Ivan Vilata
Deliverable nature:	Report (R)
Dissemination level:	Public (PU)
Contractual Delivery Date:	15/11/2013
Actual Delivery Date	20/09/2013
Suggested Readers:	Project partners
Number of pages:	28
Keywords:	Management, testbed

Authors:	Marc Aymerich, Axel Neumann, Ivan Vilata, Lorena Merino (Pangea) Bart Braem (iMinds), Leandro Navarro (UPC) Aaron Kaplan (Funkfeuer) Pau Escrich (Guifi), Javier Jimenez (Guifi)
Peer review:	Leandro Navarro, Llorenç Cerdà (UPC)

Abstract

This document presents an update of the operation and support guides for the Confine testbeds done during the second year of the project. It builds on the work reported in D3.1 during the first year of the project.

Table of Contents

Figures

Tables

1. Introduction

1.1. Contents of the deliverable

This document presents an update of the operation and support guides for the Confine testbeds done during the second year of the project. It builds on the work reported in D3.1 during the first year of the project.

The description of work document for Confine mentions this deliverable as instrumental to reflect progress in the following items:

> *"Objective 2. Provision of an experimental facility to the research community to support experimentally-driven research on these community networks"*
>
> *"T3.1 Experimental facility provision (management of the testbed) (from M06 to M48): This task is responsible for the provision of a continuously operating testbed and the provision of the specified tools, services and features to allow users carrying out experimental research. This task includes the administration of the testbed, monitoring of services and traffic."*

The description of work in Confine describes this deliverable as:

> In T3.1: "*The first results of task T3.1 will be reported in D3.1 (M12), which will describe the initial plan for operation of the experimental facility based on the integration of the different results of WP2 and will also include the license and conditions of usage for experimenters and the structure for the governance of the testbed. The following deliverables D3.2 (M24), and D3.4 (M36), be testbed management guides which include the description of the extension and integration of new features in the experimental facility made during in each reporting period.*"
>
> In Objective 2: "*Operation of the testbed, including management of the testbed (T3.1 and D3.1, D3.2, D3.4)*"

This document comprises the following topics:

- An updated operation guide, including monitoring and management aspects to keep the testbed operational.
- An updated usage guide, that experimenters can use to prepare and run experiments.
- A description of the support procedures, used to help experimenters in using the testbed.
- A description of the deployment of the testbed, and its expansion during year 2.

1.2. Relationship to other CONFINE deliverables

D2.1 Initial system software and services of the testbed – M12: A report that describes the software to construct the testbed developed during the first year.

D2.2 Initial software system for the testbed (nodes, services) - M12: A snapshot of the developed software (D2.1) at month 12. The latest version of the software can at any time be downloaded from the publicly accessible Confine repositories (http://redmine.confine-project.eu). D2.2 contains the implementation of the software system described in D2.1. The testbed described here uses this software.

D2.3 System enhancements (Year 2) - M24: D2.3 updates D2.1 and describes the new software developments in year 2. The testbed described here uses this software.

D2.6 Implementation of federation mechanisms for community networks – M24: Describe the federation mechanisms explored in the CONFINE project which are used to interconnect the diverse community networks involved in the project. D3.2 uses these concepts.

D3.1 Operation and support guides of the testbed – M12: The CONFINE project deploys its testbed for community networks called **Community-Lab**[1], and a virtual local testbed (VCT), which are based on the CONFINE testbed software system developed in WP2 and presented in D2.1 and D2.3. D3.1 therefore describes the testbed that instances the CONFINE testbed software system described in D2.1. D3.2 builds and refines D3.1.

D4.1 Experimental research on testbed for community networks (year1) - M12: D4.1 reports on experimentally driven research that was carried out to support the development of the CONFINE testbed software system. The interaction between WP2 and WP4 was bidirectional. The development of the testbed required looking at options beyond the limits of the current state-of-the-art. Work of WP4 also comprised a review of research topics relevant for community networks in order to identify use cases to be taken into account in the architecture and design of the CONFINE testbed software system. D4.1 therefore contains research work that contributed to the development of the CONFINE testbed software system.

D4.8 Tools for experimental research (Year 2) – M24. Similarly to D4.1 it reports on the research activities in the second year of the project. Several activities have used the Confine testbed.

D5.1 Dissemination, training, standardization activities in year 1 – M12: This deliverable reports on the interactions the CONFINE project had with different stakeholders in many kinds of events. While on one hand the CONFINE project was communicated to third parties, CONFINE also received valuable external feedback, which also contributed to the development of the CONFINE testbed software system reported in D2.1 and D2.3.

D5.4 Dissemination, training, standardization activities in year 2 – M24: Similarly to D5.1 it reports on the dissemination, training and standardization activities in the second year of the project.

[1] http://community-lab.net/

D5.8 Draft Exploitation Plan – M24: Describes a first draft of the exploitation plan for the testbeds developed and operated by the project.

2. Operation guide

2.1. Operation of the testbed

The Redmine project management web application[2] has been used for coordinating most of the development efforts of the CONFINE project. This service has been maintained operational since the project inception but this last year we have started using its issue tracking system for reporting and tracking all kinds of software bugs.

An e-mail address (support@community-lab.net) has been put into place in order to allow researchers and testbed users to have direct support contact with the testbed operators. The users mailing list (users@lists.community-lab.net) also has been used for researchers to discuss the doubts and problems they may be facing.

Finally rigourous backups of all servers and services like *panel.community-lab.net* or *redmine.confine-project.eu* are being maintained.

2.2. Controller and management tools for the testbed

An instance of the *confine-controller* software package has been deployed for the Community-Lab testbed at *panel.community-lab.net*. This server is the entry point for testbed users. It allows node operators to register their nodes, build and download a custom node firmware. Researchers use it for creating slices, and testbed operators can use it for management operations. This server is being maintained rigorously updated with the latest releases of the *confine-controller*[3].

Following a more detailed description of some of the work done in terms of controller and testbed management:

a) In order to ease the management of a large scale testbed a centralized management application has been developed and integrated within *confine-controller*. This application allows the execution of tasks on multiple nodes at once. Since connectivity issues or other kinds of node downtime are very frequent in large testbeds this application takes care of the tedious work of retrying a task execution when a given node comes back online.

Figure 1 illustrates how a task (operation) is created and Figure 2 shows the execution state of another operation.

b) An issue tracking system has been developed and put into production in the context of the Community-Lab testbed. Since the issue tracking system is integrated with the *confine-controller* it uses its internal users and groups therefore making it automatically available to every user of the testbed. The main goal of this system is to connect researchers, testbed operators and technicians enabling them to effectively communicate and track the issues which come up during testbed lifetime operation. They are now able to report problems directly to the person that would be able to solve them, for example

[2] http://www.redmine.org/
[3] https://wiki.confine-project.eu/soft:server-release-notes

reporting node malfunction to technicians or asking support questions to testbed operators. Figure 6 and Figure 5 are screenshots of real tickets.

c) *Confine-controller* also integrates a couple of monitoring tools designed for quickly detecting failures of testbed components.

On one hand a ping application integrated within *confine-controller* performs periodic measurements of the networked devices' latency and packet loss, and also provides nice interactive graphs for data visualization of these measurements as shown in Figure 3, easing the troubleshooting of network related issues.

On the other hand a state retrieval application performs monitoring of the state of testbed nodes which is published through their API. This application maintains a set of useful metrics to help the early detection of node malfunction. Some of these metrics can be seen on Figure 2, like "last seen", "last contact", "last try", "last change" or "current state".

d) *Confine-controller* requires to send multiple types of alerts (for example alert researchers that their slice is about to expire or warn technicians that some of their nodes are in offline state for a long period of time). A mini-framework for defining such alerts has been developed as part of the *confine-controller* package, easing the management of existing alerts and the creation of new ones.

Additionally the need for sending one-time notifications to a subset of testbed user has led to the development and deployment of a feature that enables sending e-mails to a bunch of selected users.

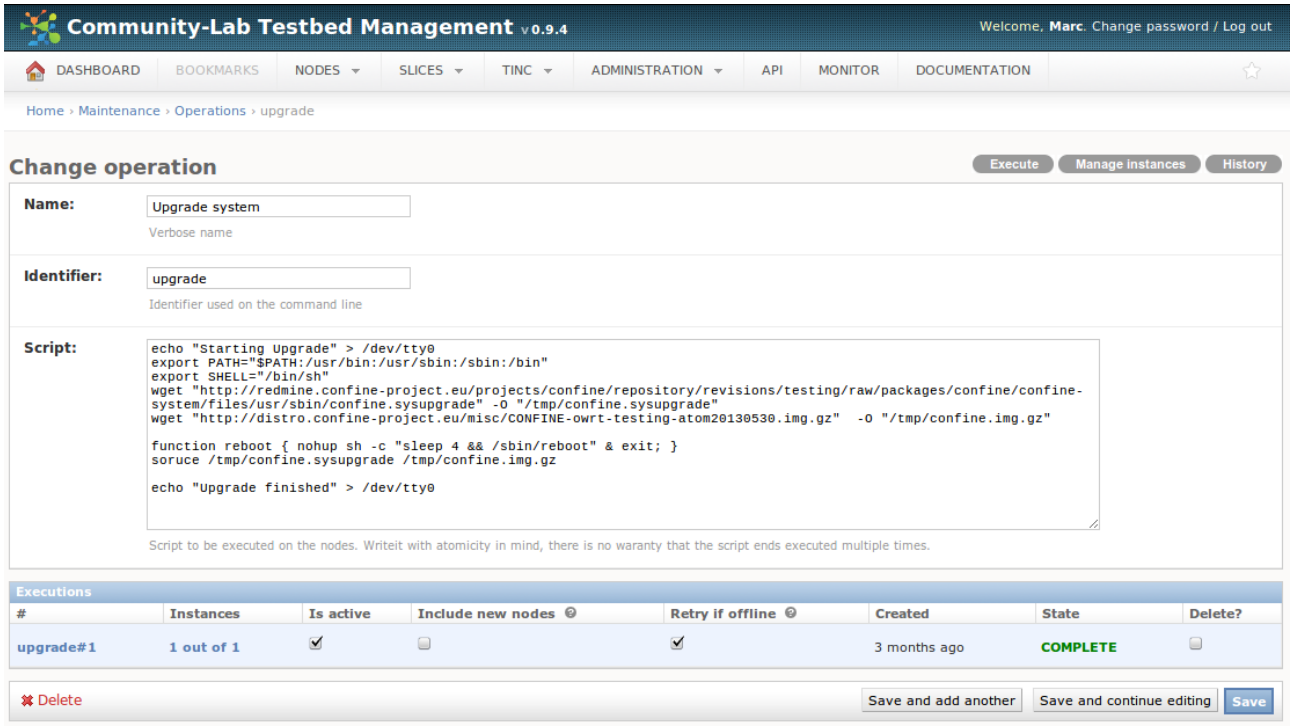

Figure 1: Maintenance operation

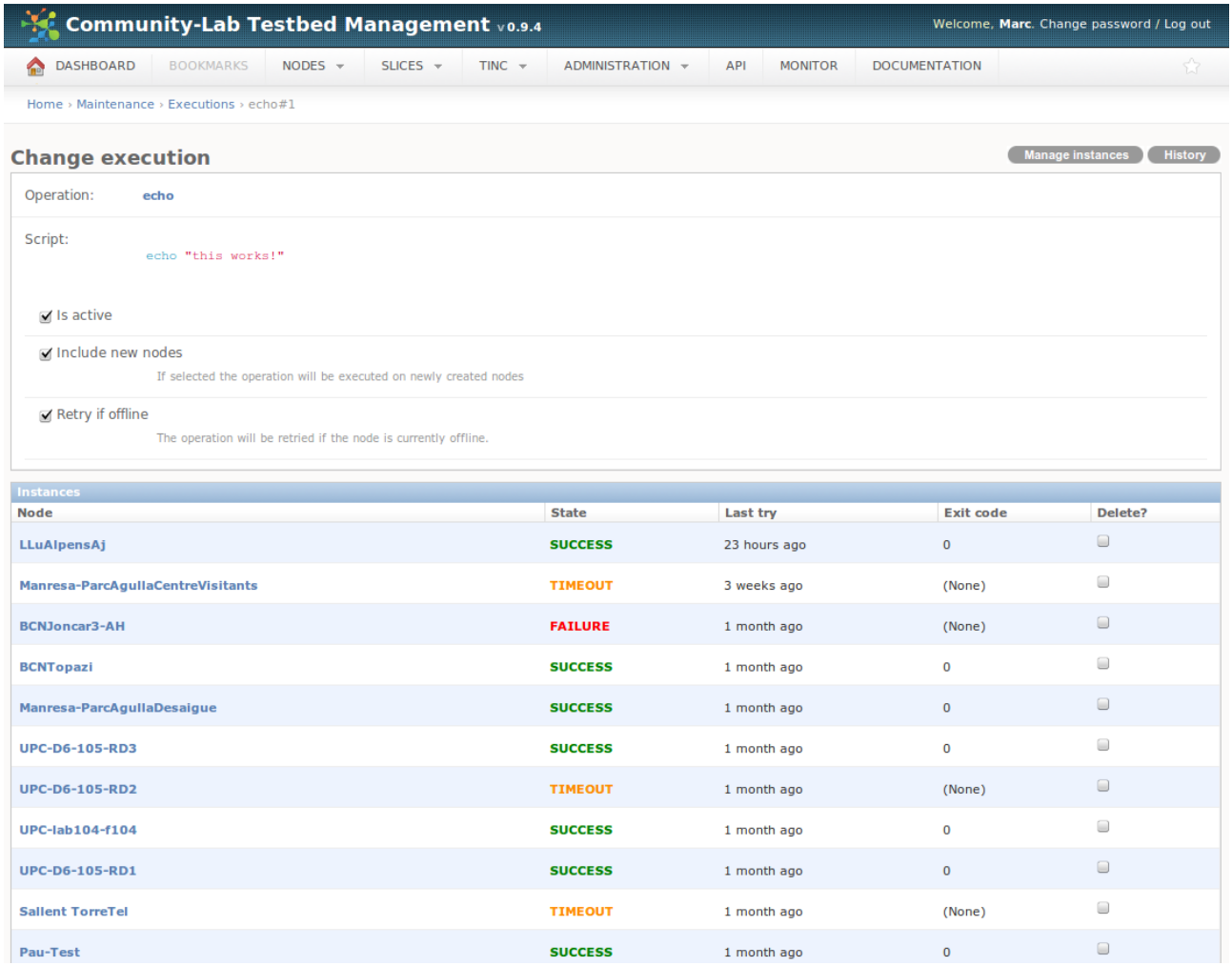

Figure 2: Maintenance execution

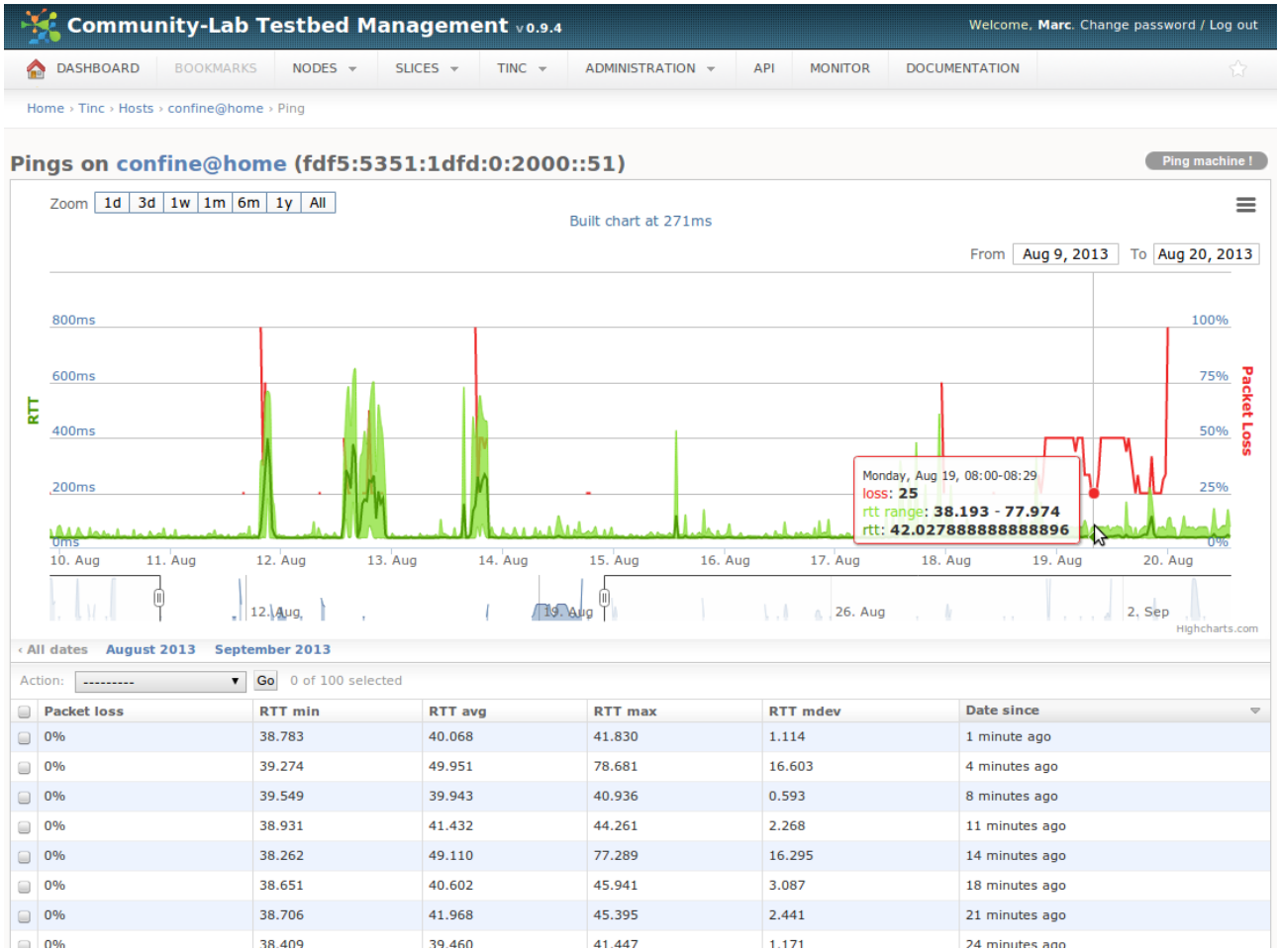

Figure 3: Pings

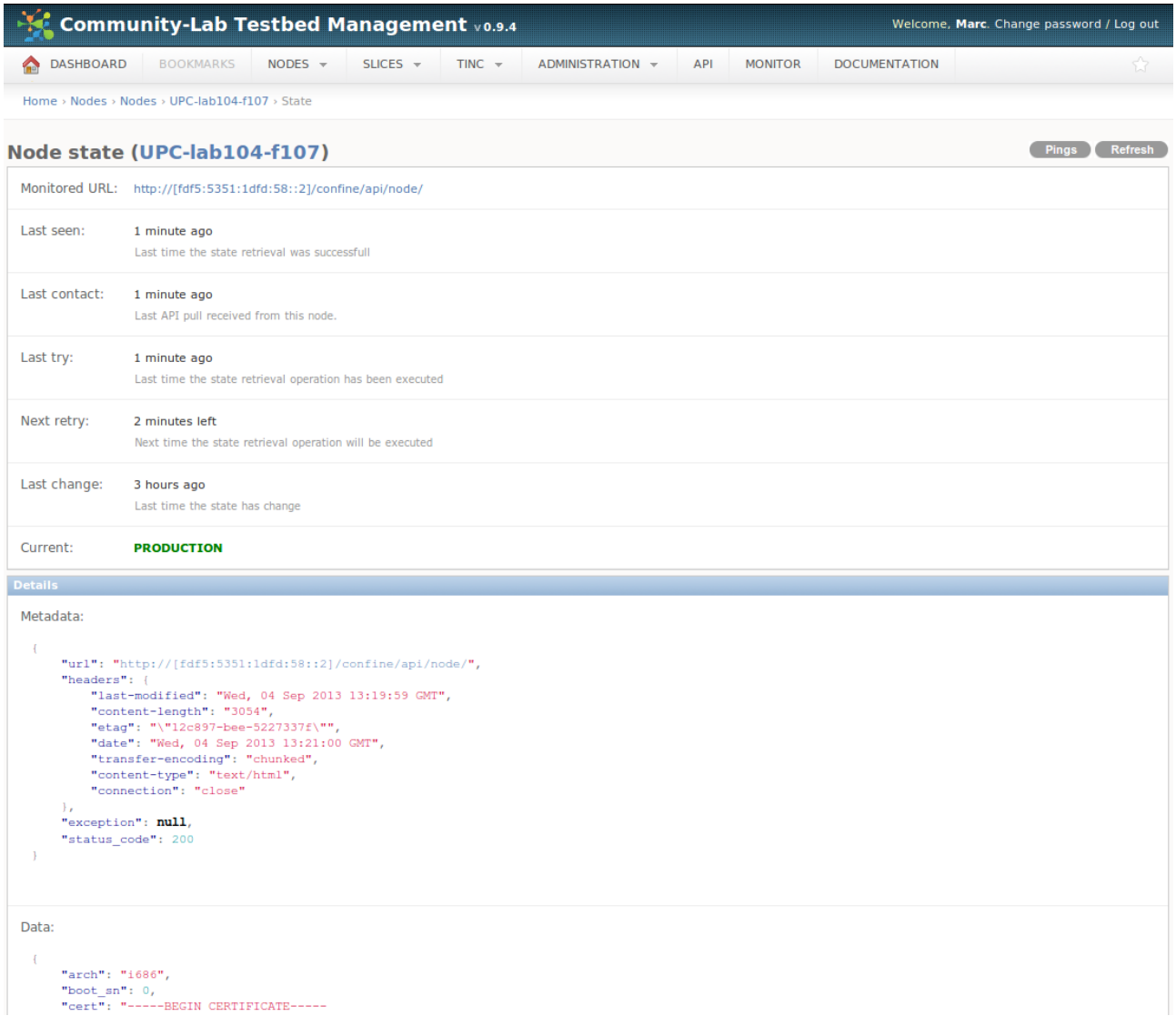

Figure 4: Node state

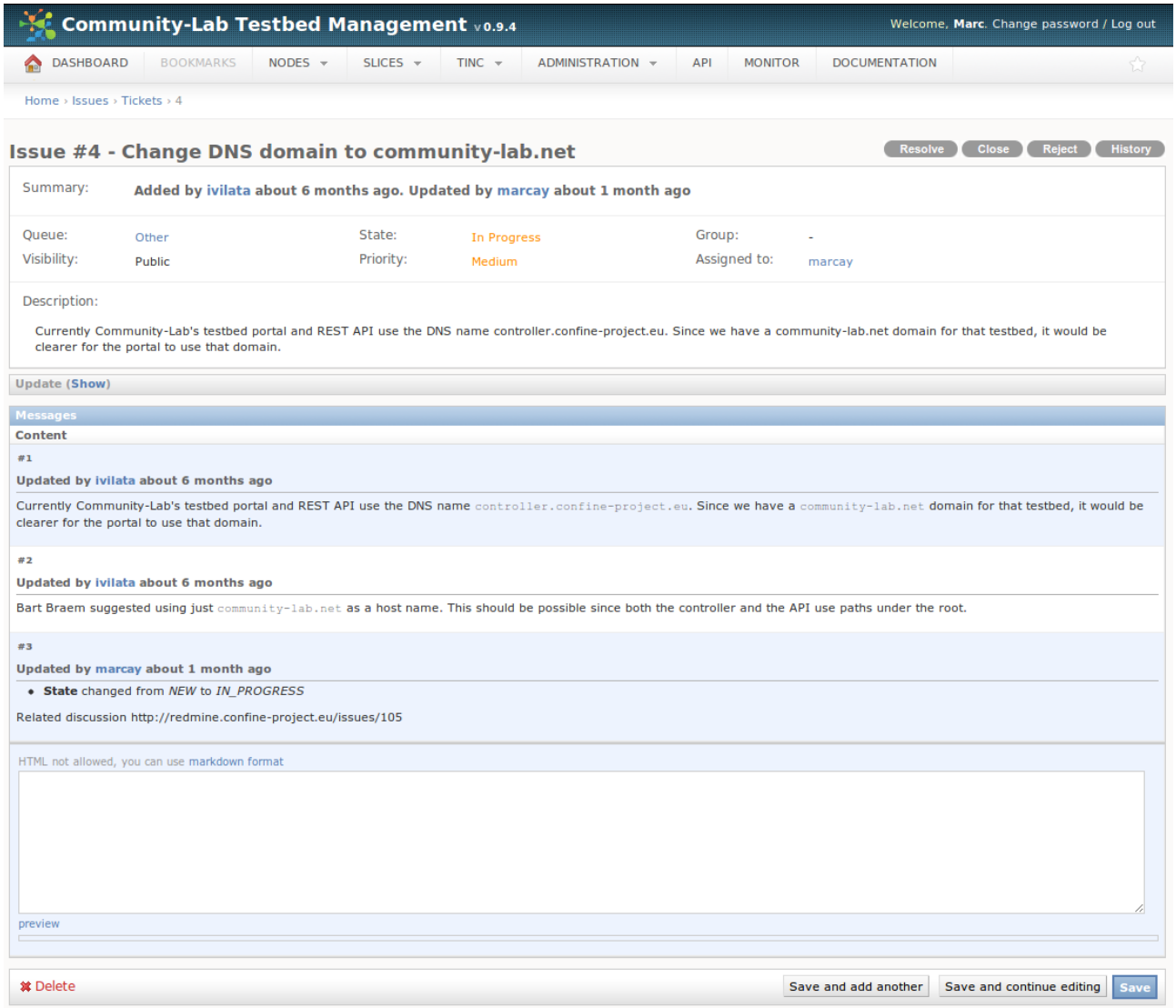

Figure 5: Ticket change

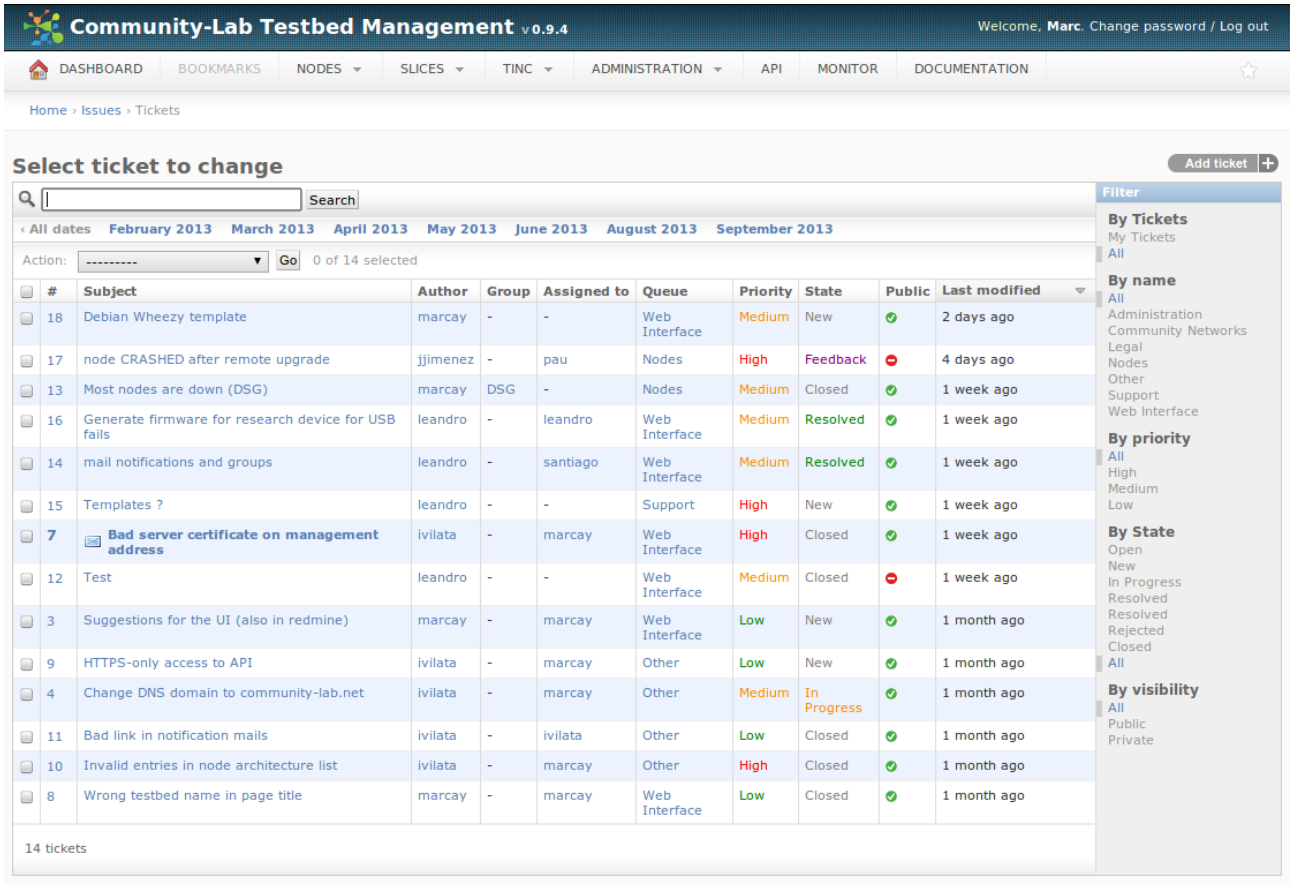

Figure 6: Ticket change list

2.3. Monitoring the underlying community network and FEDERICA

Funkfeuer is running multiple instances of smokeping[4] at: *http://tunnel.confine.funkfeuer.at/cgi-bin/smokeping.cgi?target=CONFINE_nodes*

Smokeping is a standard tool used at small and large Internet Service Providers (ISPs) for monitoring latency and uptime of servers, devices, switches and network components.

Running smokeping allowed us to monitor the availability of FEDERICA as seen from multiple perspectives. Thus, this ping “looking-glass” monitoring allowed us to pinpoint network network outages in FEDERICA.

[4] http://oss.oetiker.ch/smokeping/

In year 3 we plan to expand the smokeping service to multiple other community networks.

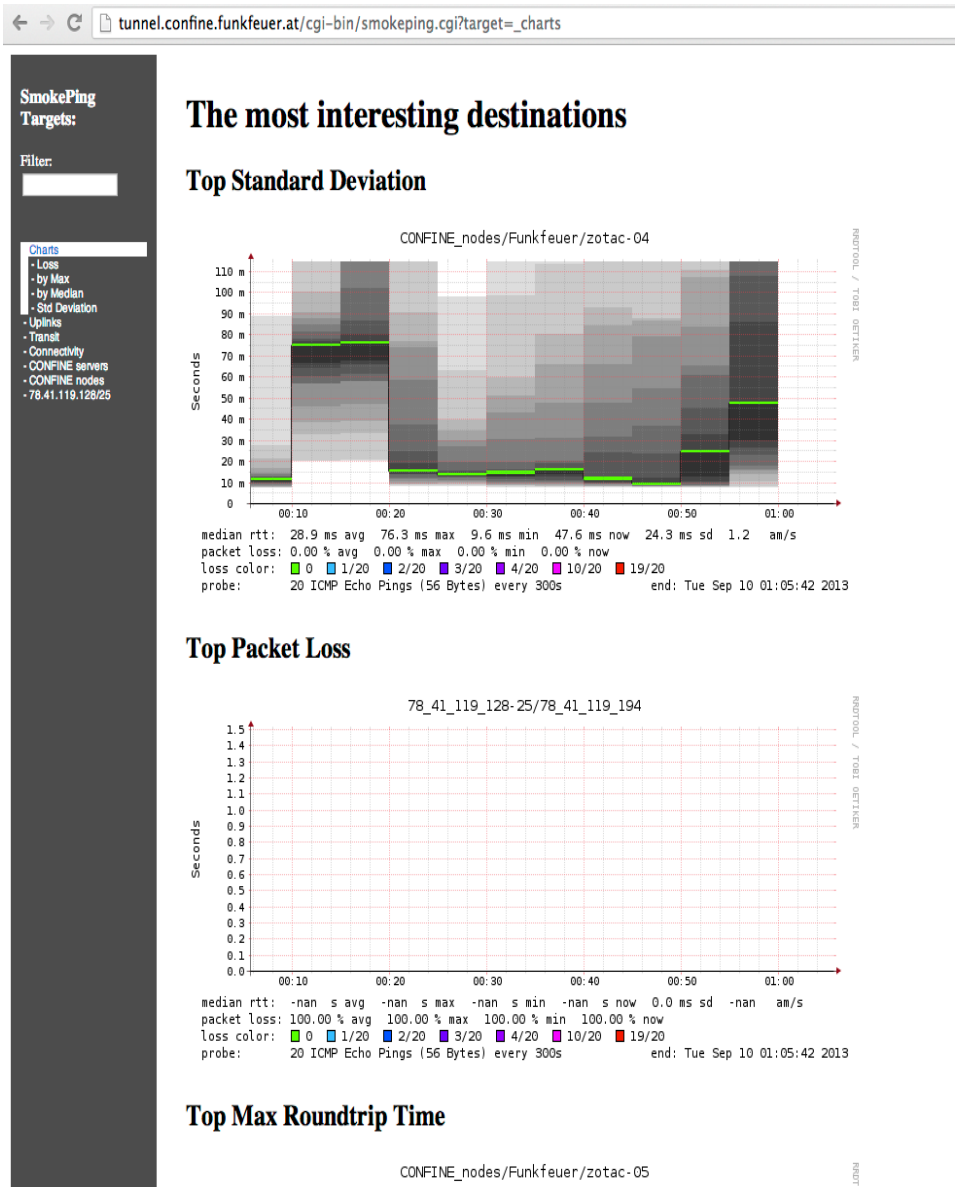

Figure 7: Example of a smokeping screenshot

3. Usage guide and documentation

Several manuals have been written to help CONFINE and Community-Lab users, developers and administrators. The following ones cover several aspects of testbed usage:

- VCT and Community-Lab tutorials: https://wiki.confine-project.eu/tutorials:start
- Experiment preparation: https://wiki.confine-project.eu/soft:node-system-bare-bones#experiment_preparation
- VCT example: https://wiki.confine-project.eu/soft:node-system-bare-bones#example
- VCT container installation and usage: https://wiki.confine-project.eu/soft:vct-container
- Node installation: https://wiki.confine-project.eu/testbeds:addnode
- Node upgrade: https://wiki.confine-project.eu/soft:node-upgrade
- Server User Manual: https://wiki.confine-project.eu/soft:server-user-manual

Other documents like the following one cover development aspects:

- https://wiki.confine-project.eu/soft:howto-compile-confine

Besides that, a user's guide has been developed that shows all the steps necessary to perform the main operations with the testbed either as a researcher, technician or group administrator.

- User's guide: https://wiki.confine-project.eu/usage:start

Similar guides shall be written for developers and testbed administrators based on the experience of developing and operating Community-Lab:

- Developer's guide: https://wiki.confine-project.eu/devel:start
- Testbed administrator's guide: https://wiki.confine-project.eu/admin:start

4. Support service

The support infrastructure[5] consists of different mailing lists, guides, wiki and a ticket system.

There is a mailing list[6] where Community-Lab users can subscribe to discuss topics and also get help and support from other users, testbed operators and developers.

Some documents and guides are available on the support wiki[7] to help users join the community, start using the testbed, run experiments, etc.

A ticket system[8] is available to report and ask assistance for problems operating the Community-Lab testbed or with nodes. Bugs and special features needed can be reported on CONFINE's Redmine site[9].

Finally, for the participant researchers of the Open Calls there are a staff member assigned to each of them for advising and supporting.

[5] https://wiki.confine-project.eu/usage:support
[6] http://lists.confine-project.eu/mailman/listinfo/users-community-lab
[7] http://wiki.confine-project.eu/
[8] https://panel.community-lab.net/admin/issues/ticket/
[9] http://redmine.confine-project.eu/projects/confine/issues

5. Testbed deployment

5.1. Node temperature stress-testing for outdoor nodes

As CONFINE nodes are deployed in open areas there is always danger of direct sunlight leading to extreme high temperatures in a node's casing. As the reliability of tests are directly correlated with the stability of a node, we have to be sure the expected maximum temperature of the case and hardware keeps below the listed safe operational temperature for the contained hardware.

As the testbeds associated with the CONFINE project cover multiple geographical and climatological areas, the obtained results from this test are probably only valid for areas with a climate more or less equal to that of Belgium. However, due to the simplicity of the test setup, it is very easy to perform similar tests on other locations.

5.1.1. Test setup

Before deploying a local testbed, one should be sure the selected casing and hardware is able to withstand the local weather conditions throughout the year. The main dangers for hardware instability arise from extreme temperatures (both high and low) and humidity causing condensation. The test was performed during a heatwave in August 2013 at the University of Antwerp, Belgium. The maximum temperature reached was 33.9 degrees Celsius (measured in a Stevenson screen). The node used for the experiment consisted out of an ALIX 3D board and a MikroTik RouterBOARD 951 (see Figure 8).

Figure 8: Hardware used for the temperature monitoring test setup: ALIX 3D (left) and MikroTik RouterBOARD 951 (right).

In order to power the ALIX, the RouterBOARD 951 was modified so it would pass through the PoE power received on port 1 to all other Ethernet ports. Due to the simplicity of our setup, this modification consisted out of connecting pins 4 and 5 from all ports with each other (V+) and soldering a wire between pins 7 and 8 of all ports (V0) (see Figure 9). Although this modification is very simple, caution has to be made as it is only allowed to power the device via one port.

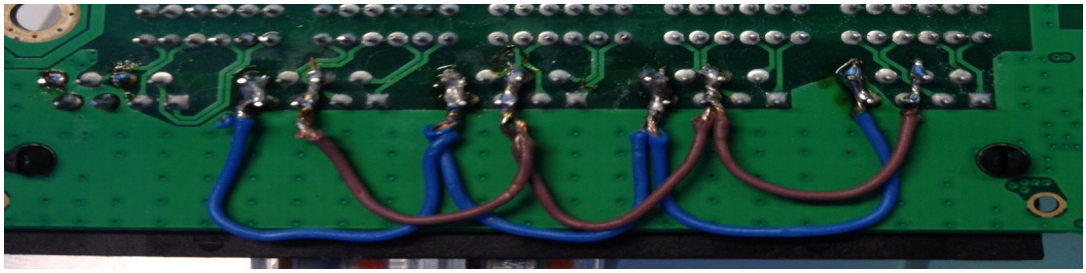

Figure 9: Passing the received power from port 0 to all other ports.

For each measurement, two temperatures were actually logged: the current temperature of the CPU and the value of a temperature sensor available on the ALIX board. Both values can be read from the kernel by reading the value of */sys/bus/i2c/devices/0-004c/temp1_input* (board) or */sys/bus/i2c/devices/0-004c/temp2_input* (CPU). Measurements were performed every 5 minutes and in order to have reference temperatures, a second ALIX 3D was placed indoors and read out simultaneously with the outdoor node.

On the hottest day, we also stress-tested the setup by executing a CPU intensive program on both ALIX boards (100% CPU utilisation) during a timespan of about 1.5 hours. The test was aborted after this time as the temperatures stabilised after about 30 minutes and remained stable during the subsequent hour.

5.1.2. Test results

The temperature values were written to an RRD file which allowed us to easily generate daily (Figure 10), weekly and monthly (Figure 11) graphs. For these graphs, the maximum over a period of 5 minutes, 1 hour and 1 day respectively was taken into account.

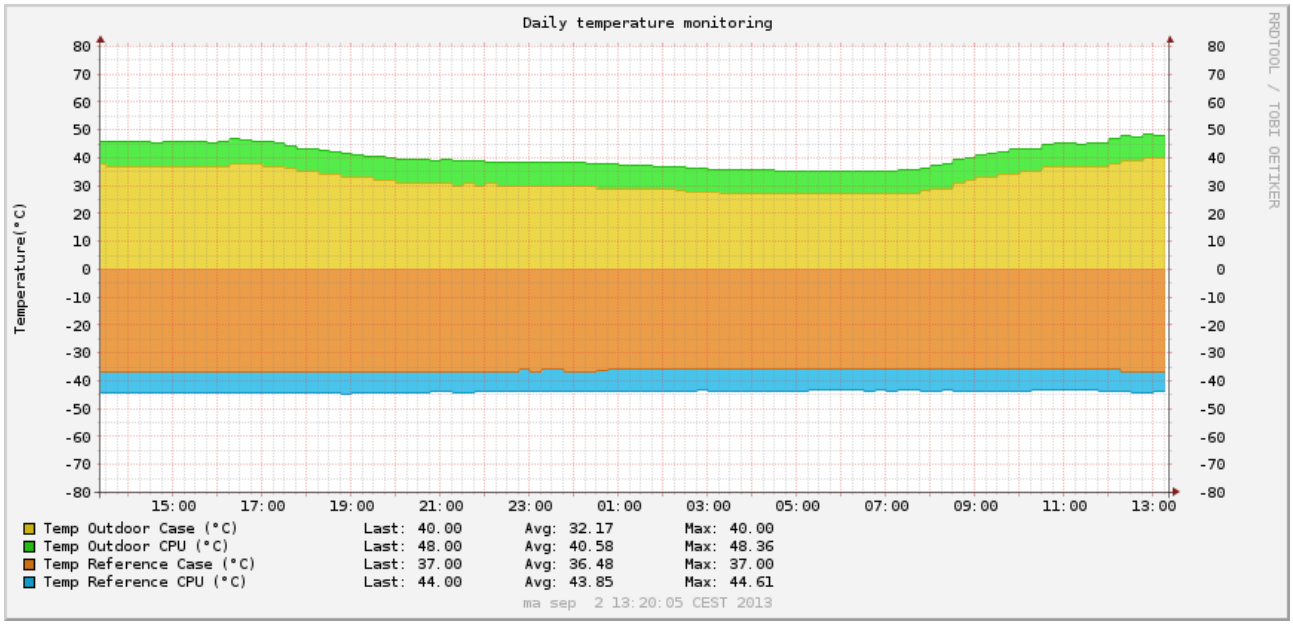

Figure 10: Daily graph

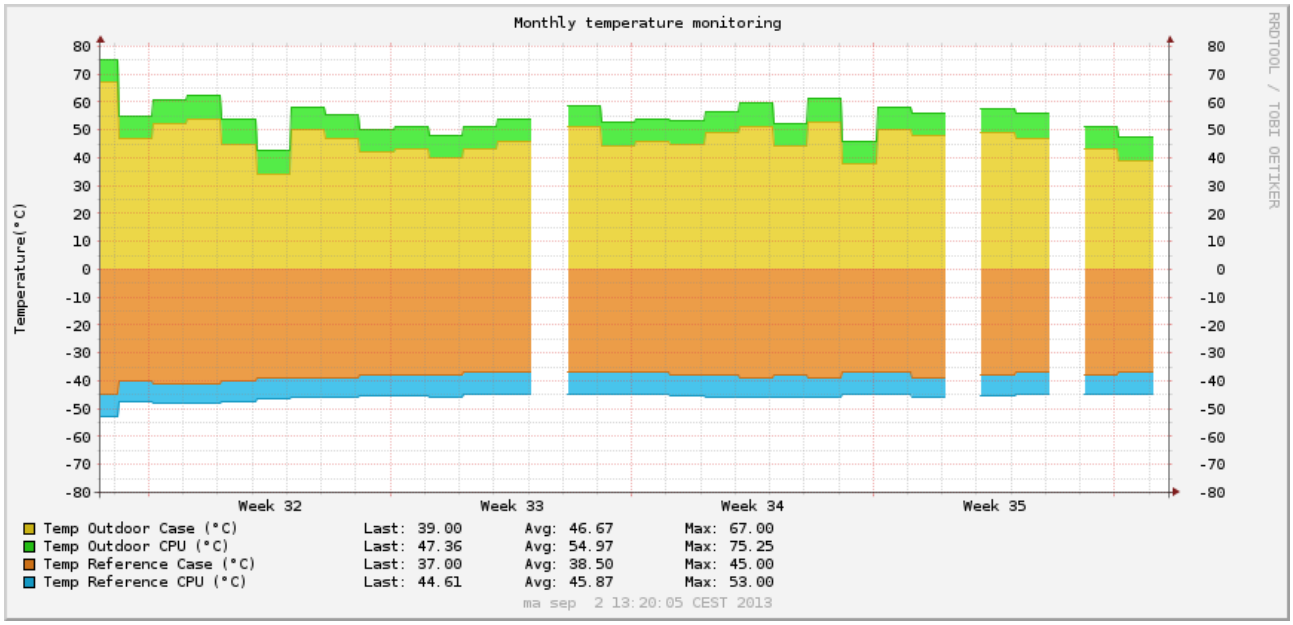

Figure 11: Monthly graph. The absence of a maximum on certain days are due to short internal networking problems.

The first thing to notice is the difference between the board and CPU temperature remained the same during the complete test. The measured temperature of the CPU is always about 8 degrees Celsius higher compared to the temperature measured from the on-board sensor.

The maximum reached temperature of the CPU (AMD Geode LX800) was 75.25 degrees Celsius, well below the listed maximum operating temperature for this processor type (85 degrees Celsius). This means the extraordinary high temperatures we encountered in our region during the month of August do not impose any particular problem for the testbed's hardware or node layout.

5.1.3. Follow-up tests

Due to the simplicity of this setup, it should not be difficult to monitor the temperature of a node on a regular basis, especially as we do currently not have any insight on how the hardware is going to behave when 'extreme' low temperatures are reached.

However, a drawback of this monitoring mechanism exists in the lack of a way to monitor humidity in the enclosure box. This might become an issue especially on colder days to keep an eye on.

5.2. Deployment of nodes

In this second year the efforts on the deployment of research nodes have been specially focused on the indoor type device since all the accepted proposals of the first Open Call except one do not require link layer control in their experiments.

These nodes do not have operational Wi-Fi devices but they are more powerful (in terms of CPU, memory and storage) than the outdoor nodes deployed during the first year. Thus, they suit better for experimentation in the application layer.

The deployment has been mainly achieved through the subproject named confine@home, which promotes the spread of CONFINE nodes among community network participants. The nodes are purchased, mounted and configured by the CONFINE project and the participants are in charge of the collocation and the maintenance. The CONFINE project will keep the ownership of these nodes during its duration. Afterwards they might by donated to the participants (not yet decided).

The confine@home participants have to connect the research device to their community node through a wired interface (RJ45 Ethernet). Once it is done the node will appear as ready to be used for experimentation in the Community-Lab web controller. Using the web interface the user can see the status of the node, statistics, experiments running, etc.

The next guide has been published to let the confine@home users know how to install and configure the nodes themselves.

https://wiki.confine-project.eu/testbeds:addnode

The following table presents the current status of the deployment, as of September 2013[10] (82 nodes):

Name	Arch.	Group	Name	Arch.	Group
AWMN-dem-chem	i686	AWMN	UPC-Omega	i586	DSG
Awmn-dem-iit	i586	AWMN	UPC-C6	i586	DSG
AWMN-dem-ims	i586	AWMN	UPC-D1	i586	DSG
AWMN-CF-djk604	i686	AWMN	RD_Commel_FF	i686	DSG

[10]This information can be obtained from the controller web page https://panel.community-lab.net/admin/ under Nodes, select Filter, By nodes, All.

Name	Arch.	Group	Name	Arch.	Group
AWMN-CF-Wolfpack-alix	i586	AWMN	RD_FF	i686	DSG
AWMN-DA-Town Hall	i686	AWMN	UPC-lab104-f104	i686	DSG
AWMN-DA-MEC	i686	AWMN	SantFruitosDiposit	i686	Guifi.net
AWMN-CF-7bpm	i686	AWMN	SBDAlfonsXIII	i686	Guifi.net
AWMN-CF-7bpm-3	i686	AWMN	Manresa-ParcAgullaCV	i586	Guifi.net
AWMN-CF-Infolex	i686	AWMN	BCNJoncar3-AH	i686	Guifi.net
AWMN-CF-ipduh	i686	AWMN	Manresa-ParcAgullaDes	i586	Guifi.net
AWMN-CF-Ymitos	i686	AWMN	BCNTopazi	i686	Guifi.net
AWMN-DA-KAPI	i686	AWMN	Sallent TorreTel	i686	Guifi.net
AWMN-DA-GYM	i686	AWMN	Pau-Test	i686	Guifi.net
AWMN-CF-7bpm-2	i686	AWMN	Manresa-UPC	i586	Guifi.net
AWMN-CF-Wolfpack	i686	AWMN	Vic-ldalmau	x86_64	Guifi.net
UPC-D6-105-RD3	i686	DSG	Vic-BarriOsona	x86_64	Guifi.net
UPC-D6-105-RD2	i686	DSG	LLUalpensAj	i686	Guifi.net
UPC-D6-105-RD1	i686	DSG	LLUolostAj	x86_64	Guifi.net
UPC-CSTF-EETAC01	i686	DSG	LLUsbgTorre	i686	Guifi.net
UPC-CASTF1	i586	DSG	LLUperafitaPriona	i686	Guifi.net
UPC-lab104-dani	i686	DSG	StFruitos-Aigues	i686	Guifi.net
UPC-lab104-demo4	i686	DSG	Santpedor-Aigues	i686	Guifi.net
UPC-lab104-demo3	i686	DSG	Manresa-Passeig	i686	Guifi.net
UPC-lab104-demo2	i686	DSG	Stack_A2	i686	Guifi.net
UPC-lab104-demo1	i686	DSG	Stack_A1	i686	Guifi.net
UPC-lab104-f107	i686	DSG	Manresa-AdeM	i586	Guifi.net
UPC-lab104-f105	i686	DSG	BCNBenlliure6	i686	Guifi.net
UPC-lab104-f106	i686	DSG	BCNPalafolls33	i586	Guifi.net
UPC-lab104-f108	i686	DSG	HW-ermita11	i686	Guifi.net
UPC-lab104-f001	i686	DSG	BCNAlcolea46	i686	Guifi.net
UPC-C6-102	i686	DSG	BCNJocsFlorals	i686	Guifi.net
UPC-lab104-f003	i686	DSG	Vic-elSerrat	i686	Guifi.net

Name	Arch.	Group	Name	Arch.	Group
UPC-lab104-f002	i686	DSG	BCNSjmalta-outdoor	i686	Guifi.net
UPC-lab104-f103	i686	DSG	Manresa-PF	i586	Guifi.net
UPC-lab104-f102	i686	DSG	Manresa-SantaCaterina	i586	Guifi.net
UPC-lab104-f101	i686	DSG	Manresa-JoanXXIII	i586	Guifi.net
UPC-C6E206DSG	i686	DSG	Manresa-PTST	i586	Guifi.net
UPC-D6-105	i686	DSG	Manresa-PuigBerenguer	i586	Guifi.net
UPC-H	i586	DSG	Manresa-SantIgnasi	i586	Guifi.net
UPC-A1	i586	DSG	UPC-Vertex-Pangea	i686	Pangea

The CONFINE system is compatible with most of the computers based on an Intel x86 CPU. The ones selected for the subproject confine@home have the following characreristics:

- CPU Intel Atom N2600
- 2GB DD3 RAM
- 128GB SSD SATA3
- 2x Gigalan Ethernet Port
- 1x WiFi Atheros card (not used)

There is no fan or other mobile components.

iMinds has also started collaborating with the local community network, in order to be able to generate data from this network. Node hardware has been bought for one location at their campus and helped install hardware for two other locations at our campus.

In November 2012, iMinds contacted the local community network Wireless Antwerpen to discuss collaboration. This would allow iMinds to study the local context of community networking, which is very different given the Belgian economic situation and its broadband penetration. Moreover, it would help iMinds staff get a hands on demonstration of community networking. An informal agreement was reached, in which iMinds purchased node hardware to actively participate in Wireless Antwerpen and helped install these nodes on one of the roofs of its University of Antwerp campus. With this informal collaboration, iMinds researchers have been able to more profoundly study community networks.

As a result, iMinds developed a community network mapper which will be presented during the International Summit for Community Wireless Networks in October 2013 in Berlin, see deliverable D4.8 for more information.

Another interesting collaboration involves the logging of network traffic over the Wireless Antwerpen community network (with respect for personally identifying information). This will allow iMinds to build models of community network traffic, which perfectly integrate in the multiple layer approach to community networking as outlined in deliverable D4.2.

Finally, to strengthen the collaboration and to really experience community networking, in the spring of 2013 iMinds has provided locations for two more nodes on its University of Antwerp campus. The iMinds researchers actively contributed during the installation of the hardware, learning about all different aspects involved in installing a wireless community network node.

5.3. Interconnection

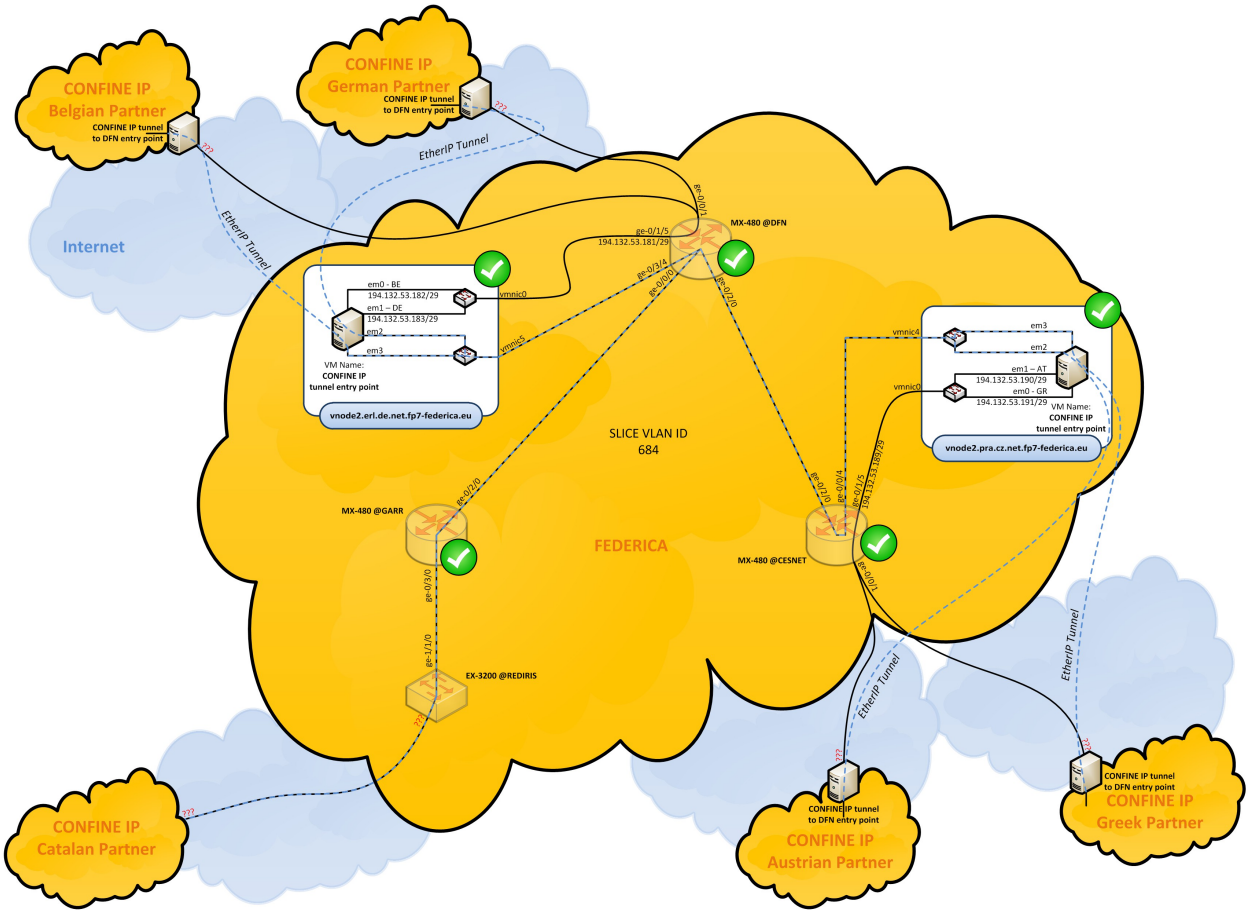

Figure 12: CONFINE partners connected over FEDERICA

As of the date of writing this document, the interconnection of the testbeds via the FEDERICA network (depicted in Figure 12) [FEDERICA] is only working reliably between iMinds, Funkfeuer, FKIE and AWMN. The interconnection with Guifi, Pangea and UPC towards GARR uses a common L2 link (a VLAN over RedIris, the Spanish academic network) that has been suffering many outages in the last few months. This is mostly due to network problems which existed at the FEDERICA / GEANT network.

After multiple attempts we are moving towards interconnecting the testbeds directly via tunnels (EtherIP, tinc, OpenVPN and similar). This should at least ensure a fallback connectivity via a tunnel if the FEDERICA interconnection does not work.

5.4. CONFINE servers

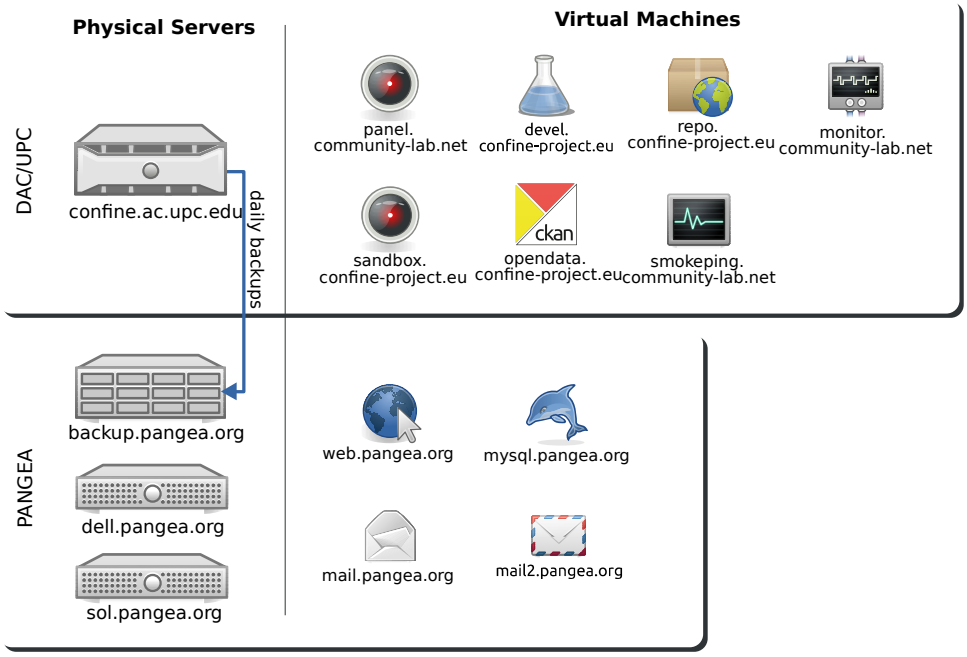

Figure 13: CONFINE server infrastructure at the UPC and Pangea

Besides the main Community-Lab and CONFINE servers located in the UPC and Pangea (see Figure 13), iMinds has a separate CONFINE controller running in its experimental testbed. This controller was installed with cooperation from Pangea, as a verification of the controller installation documentation. This controller has been installed separately as a platform for extension, with OMF [OMF] and federation.

The OMF extension is meant as a basis to allow for further testing and stabilising of the OMF code written by iMinds, to avoid interrupting the functionality of the standard CONFINE controller software which is also used by the open call participants for real experiments.

The local CONFINE controller is also being maintained to allow for federation in the future, where multiple CONFINE controllers are federated over multiple testbeds.

The Fraunhofer FKIE maintains a separate CONFINE server instance for the operation of its local testbed. The testbed is meant for the early phases of experimentation before using the larger community network testbeds. Once the server software supports federation, it is planned to federate with the other CONFINE testbed instances.

Funkfeuer purchased a server for CONFINE (*stats.funkfeuer.at*) which is being used for collecting statistics of the network, for interconnecting between community wireless networks and for spidering data, running a node DB and announcing the testbed via BGP to the internet. Should the need arise to run an extra Community-Lab server instance, Funkfeuer can do so at any moment since it uses public IP addresses for the testbed.

6. Conclusions

The work in the second year of WP3 has been focused on preparing an environment to provide a production testbed that incorporates the software developed in WP2. This implies

- Testing and integrating the software components into the production testbed for Community-Lab and as a software package to download for the VCT virtual testbed.
- Documenting the procedures to use the testbed for experiments and to expand the testbed buy adding new nodes.
- Managing the continued operation of the testbed by monitoring and acting on the different elements.
- Providing support services for all types of users.
- Upgrading the different elements as software evolves.

This deliverable illustrates the many facets and the complexity of this enterprise.

7. References

[FEDERICA] FEDERICA, Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures: http://www.fp7-federica.eu/

[OMF] The cOntrol and Management Framework: **http://mytestbed.net/**

The CONFINE project

September 2013

CONFINE-201309-D3.2-1.0